

# Aufgabenstellung: Klassifikation

Aufgabenstellung:

- Trainingsdaten: Geben sei eine Menge von Bilddaten  $\vec{x}$  mit  $\vec{x} = [x_1, x_2, \dots, x_t]$
- Klassen: Die Bilder lassen sich in  $k$  - Klassen einteilen.
- Klassifikation: Es sollen ähnliche Bilder anhand der Trainingbilder klassifiziert werden.

# Probleme der Klassifikation

Problem:

- Bilder sind meist größer als 76800 Pixel  $\Rightarrow$  direkter Vergleich zwischen Bildern aufwendig (Dimensionsproblem)
- Keine Genrealisierung

# Lösung des Dimensionsproblems mittels PCA

Prinzip: Projektion des eigentlichen Problems auf einen Subraum, in dem die Varianzen der Klassen hervorgehoben werden.

Dimensionsreduzierung: Ergibt sich aus der Projektion  $\Rightarrow$  Im Subraum werden (meist) weniger Merkmale benötigt

# PCA im Detail

## 1. Daten-Aquirierung:

Für den Aufbau des Subraumes müssen zunächst Trainingsdaten ( $\vec{x}_M$ ) gesammelt werden.

## 2. Mittelwertbereinigung:

Damit die PCA korrekt funktioniert müssen alle Trainingsdaten Mittelwertbereinigt sein.

$$\vec{x} - \vec{\mu} \quad \text{mit} \quad (1)$$

$$\vec{\mu} = \frac{1}{M} \sum_{i=1}^M \vec{x}_M \quad (2)$$

### 3. Berechnung der Kovarianz:

Aus den mittelwertbereinigten Daten wird die Kovarianzmatrix

$$\mathbf{Q} = \mathbf{P}\mathbf{P}^T \quad \text{mit} \quad (3)$$

$$\mathbf{P} = [\vec{x}_1 - \vec{\mu}, \vec{x}_2 - \vec{\mu}, \dots, \vec{x}_M - \vec{\mu}] \quad (4)$$

berechnet.

### 4. Berechnung der Eigenwerte:

Die Eigenwerte der Kovarianzmatrix lassen sich durch folgende Gleichung bestimmen:

$$\lambda_i \vec{e}_i = \mathbf{Q} \vec{e}_i \quad \text{mit} \quad (5)$$

$\lambda_i$   $i = 1 \dots m$  Eigenwerten und

$\vec{e}_i$   $i = 1 \dots m$  Eigenvektoren

5. Dimensionsreduktion :  
Sortiere die  $\vec{e}_i$  so das gilt :

$$\lambda_1 > \lambda_2 > \dots > \lambda_m$$

Somit nimmt der Informationsgehalt mit Index  $i$  ab.  
Für die Dimensionsreduktion werden nur die ersten  $n$   
Eigenvektoren benutzt ( $n < m$ ) da in ihnen die meiste  
Information enthalten ist.

## 6. Transformationsmatrix:

Mit der Transformationsmatrix

$$\mathbf{A} = (\vec{e}_1 \dots \vec{e}_n)^T$$

lassen sich die Eingangsdaten in den Eigenraum transformieren:

$$\vec{p}_i = \mathbf{A} \cdot \vec{x}_i \quad \dim(\vec{p}_i) = n$$

## 7. Rückprojektion:

Da  $\mathbf{A}$  quadratisch und orthogonal gilt:

$$\mathbf{A}^{-1} = \mathbf{A}^T$$

Somit ergibt sich die Rücktransformation durch:

$$\vec{x}_i = \mathbf{A}^{-1} \vec{p}_i = \mathbf{A}^T \vec{p}_i$$

# Dimensionsbestimmung

- Wird  $n$  groß gewählt  
⇒ wenig Informationsverlust aber auch keine große Dimensionsreduktion
- Wird  $n$  klein gewählt  
⇒ uU. viel Informationsverlust aber nachfolgende Operationen einfacher

Automatische Wahl von  $n$  :

$$\frac{\sum_{i=1}^n \lambda_i}{\sum_{i=1}^m \lambda_i} \geq T$$

# Implizite Kovarianz

Berechnung der Kovarianzmatrix problematisch:

Bild von  $320 \times 240$  Pixeln  $\Rightarrow$  Vektor mit der Dimension 76800

Kovarianzmatrix :

$$\mathbf{Q} = \mathbf{P}\mathbf{P}^T \quad \mathbf{Q} \in M_{t \times t}$$

$\Rightarrow 76800^2 = 5,89824 * 10^9$  Elemente

$\Rightarrow$  bei einem Byte pro Element  $\approx 5,5GB$

Normalerweise hat man aber nur  $M \ll \dim(\vec{x}_i)$  Eingangsdaten und somit kann man auch höchstens  $M$  Eigenvektoren finden.

Implizite Kovarianz:

$$\tilde{\mathbf{Q}} = \mathbf{P}^T \mathbf{P} \quad \tilde{\mathbf{Q}} \in M_{M \times M}$$

Bei Beispielsweise 100 Bildern als Input:

$\Rightarrow 100^2 = 10000$  Elemente

$\Rightarrow$  bei einem Byte pro Element  $\approx 9,77kb$

# Eigenwerte und Eigenvektoren der impliziten Kovarianzmatrix

Die Eigenwerte von  $\mathbf{Q}$  und ihre korrespondierenden lassen sich aus den Eigenwerten und -vektoren von  $\tilde{\mathbf{Q}}$  berechnen:

$$\begin{aligned}\lambda_i &= \tilde{\lambda}_i \\ \vec{e}_i &= \tilde{\lambda}_i^{-\frac{1}{2}} \mathbf{P} \tilde{\vec{e}}_i\end{aligned}$$

# Klassifikation und PCA

Beispiel aus „Turk and Pentland: Eigenfaces for Recognition “

- Projektion der Klassen in den Eigenraum:

$$\vec{\Omega}_c = \mathbf{A}^T (\vec{x}_c - \vec{\mu}) \quad c = 1, \dots, k \quad (6)$$

- Bestimme maximalen Abstand zwischen Klassen:

$$\theta_l = \frac{1}{2} \max_{j,k} \{ \|\vec{\Omega}_j - \vec{\Omega}_i\| \} \quad j, i = 1, \dots, k \quad (7)$$

- Klassifikation eines neuen Bildes  $\vec{x}$ :

1. Projektion in den Eigenraum :

$$\vec{\Omega} = \mathbf{A}^T (\vec{x} - \vec{\mu}) \quad (8)$$

2. Klassenabstand bestimmen:

$$\epsilon_c = \|\vec{\Omega} - \vec{\Omega}_c\| \quad (9)$$

3. Bestimmung des Abstands zwischen Eingabe und Rückprojektion:

$$\epsilon = \|\vec{x} - \vec{x}_r\| \quad \text{mit} \quad (10)$$

$$\vec{x}_r = \mathbf{A}\vec{\Omega} + \vec{\mu} \quad (11)$$

#### 4. Klassifikation

– Falls  $\epsilon \geq \theta_l$

Input ist kein Gesicht.

– Falls  $\epsilon < \theta_l$  und  $\forall c, \epsilon_c \geq \theta_l$

Input ist ein unbekanntes Gesicht.

– Falls  $\epsilon < \theta_l$  und  $\epsilon_{c^*} = \min_c \{\epsilon_c\} < \theta_l$

Input enthält ein Gesicht von Person  $c^*$

# PCA und Neuronale Netze

Die PCA lässt sich auch über die Hebb-Regel von einem einschichtigen Perzeptronnetzwerk lernen.

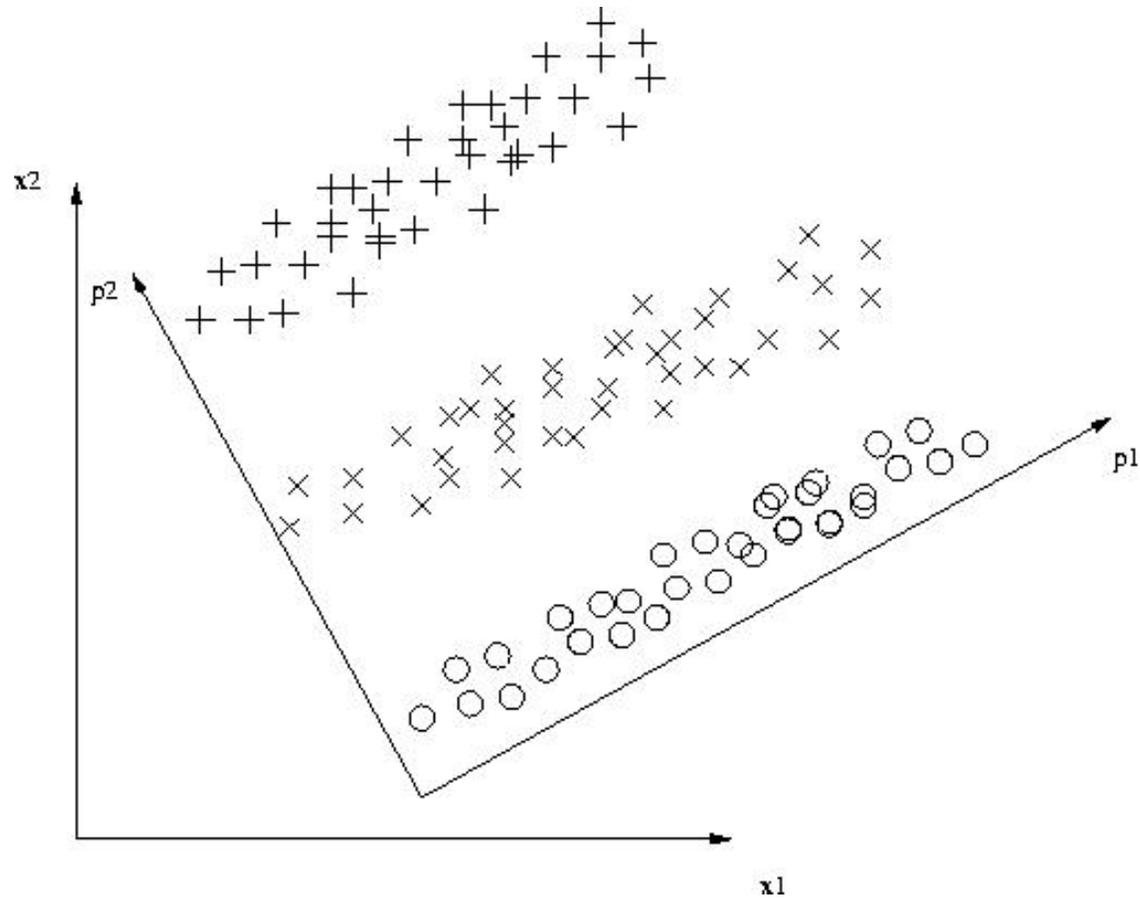
Regel von Yuille et al. :

$$\begin{aligned}\vec{y} &= \sum \omega_i \xi_j = \mathbf{w}^T \vec{x} = \vec{x}^T \mathbf{w} \\ \Delta \omega_j &= \eta (V x_j - \omega_j |w|^2)\end{aligned}$$

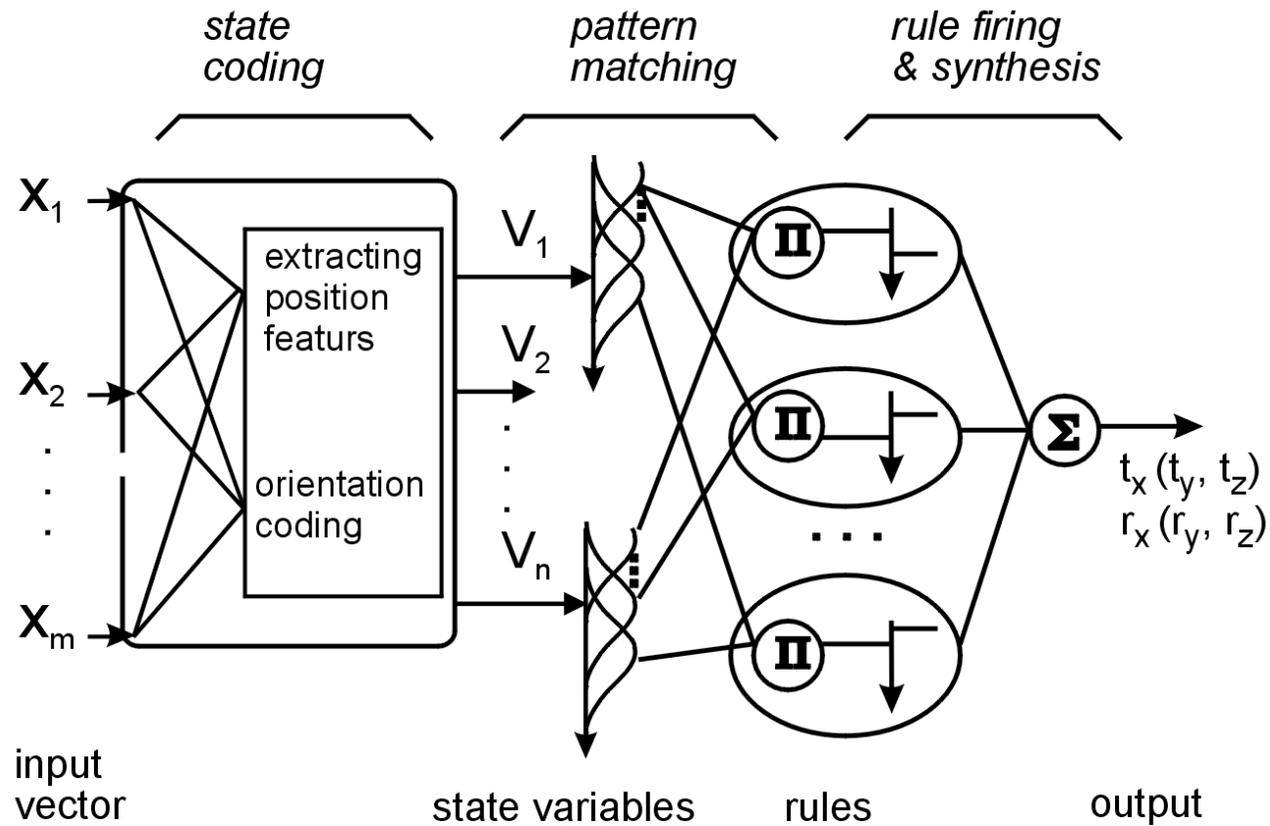
Der Gewichtsvektor  $\vec{w}$  zeigt im Konvergenzfall in die Richtung des Eigenvektors der Kovarianzmatrix mit der größten Varianz.

# Problemfälle für PCA

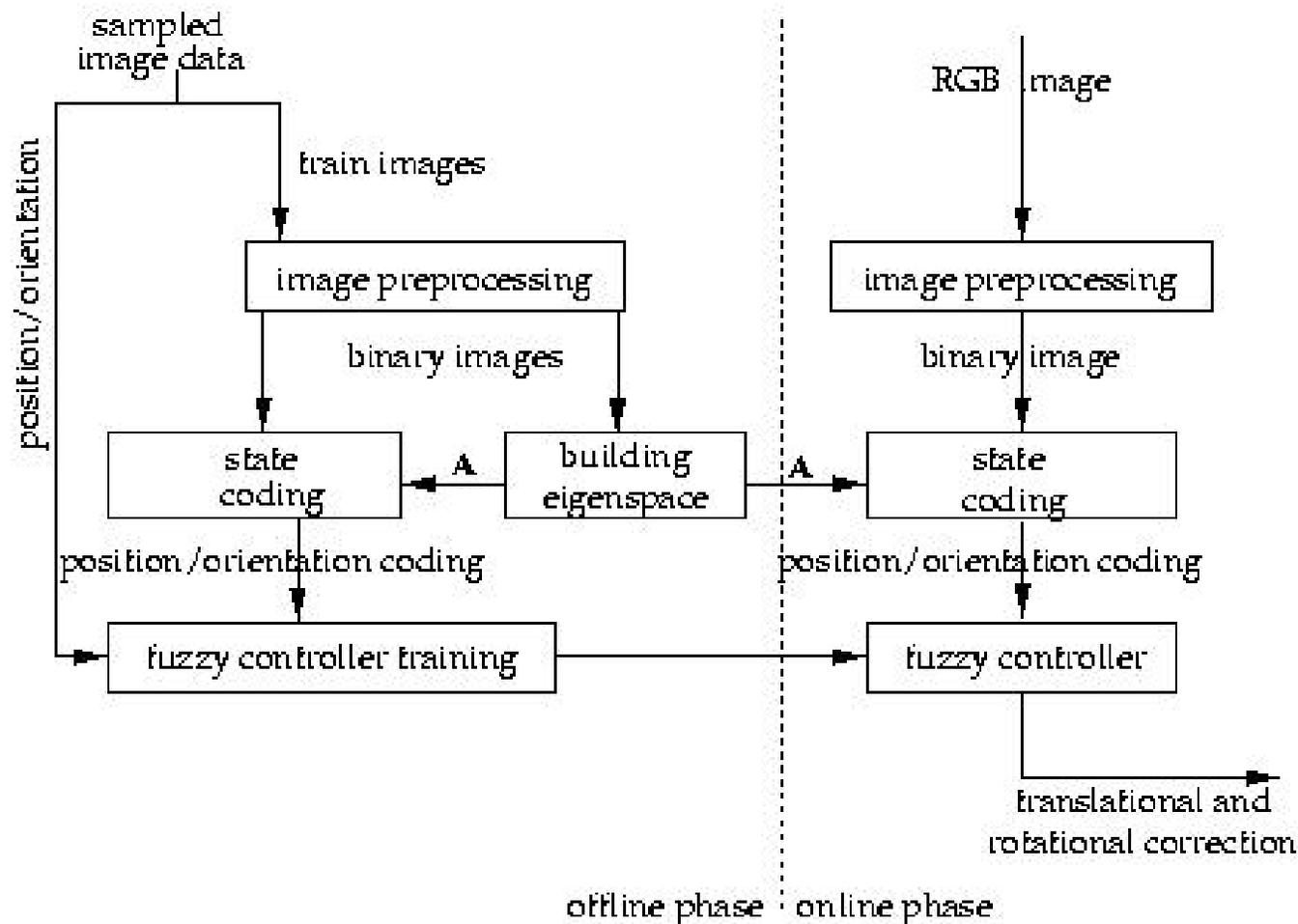
Addidas-Problem:



# Das Neuro-Fuzzy-Model



# Die Trainings- und Anwendungsphasen



# Independent Component Analysis

Folgender Sachverhalt verdeutlicht die Arbeitsweise der Independent Component Analysis (kurz ICA):

In einem Raum sind zwei Lautsprecher aufgestellt, die zwei verschiedene Tonsignale  $s_1(t)$  und  $s_2(t)$  ausgeben.

Die zwei Tonsignale werden von zwei verschiedenen Mikrofonen an verschiedenen Stellen aufgenommen.

Die beiden Mikrofone nehmen zwei unterschiedliche Mischungen  $x_1(t)$  und  $x_2(t)$  der Originaldaten auf, wobei:

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t) \quad (12)$$

$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t) \quad (13)$$

Die Faktoren  $a_{11}, a_{12}, a_{21}, a_{22}$  und die Originaldaten sind aus der Perspektive der Mikrofone unbekannt. In Vektornotation kann die obige Gleichung allgemein wie folgt geschrieben werden:

$$\vec{x}(t) = \mathbf{A} \cdot \vec{s}(t) \quad (14)$$

wobei  $\vec{x}(t), \vec{s}(t) \in R^n$  sind und  $\mathbf{A}$  eine  $n \times n$ -Matrix ist.

Falls die Originaldaten folgenden Bedingungen genügen, kann die Mischungsmatrix  $\mathbf{A}$  mit der ICA bestimmt werden:

1. Die Originaldaten müssen statistisch unabhängig sein
2. Die Originaldaten müssen stationär sein
3. Maximal eine Originalquelle darf gaußverteilt sein

Die Originaldaten können aus der inversen Mischungsmatrix  $W = A^{-1}$  der Matrix  $A$  berechnet werden:

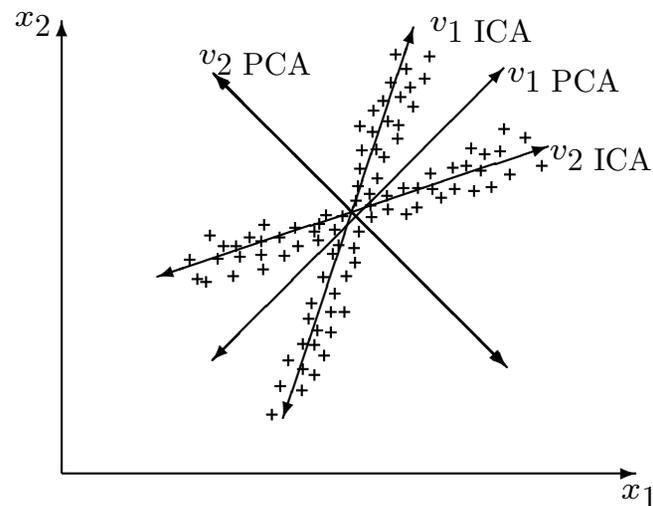
$$\mathbf{s}(t) = W \cdot \mathbf{x}(t) \quad (15)$$

Die Wiederherstellung der ursprünglichen Daten unterliegt allerdings zwei Einschränkungen:

1. Die Energie oder die Varianzen der einzelnen Originalquellen können nicht wiederhergestellt werden.
2. Die Reihenfolge der Originalsignale  $s_i(t)$  kann ebenfalls nicht rekonstruiert werden.

# Vergleich ICA mit PCA

Gezeichnet sind die Vektoren zweier korrelierter, nicht-normalverteilter Zeitserien. Die PCA projiziert auf eine Basis, deren Achsen orthogonal sind, wobei die 1. Achse in Richtung der größten Varianz zeigt. Die Achsen der ICA müssen nicht orthonormal sein, so daß die Varianz für beide Achsen maximiert werden kann. Dieses führt zu einer günstigeren Dekorrelation.



# Dimensionsreduktion mit der ICA

Bei der PCA ist die Reihenfolge der zu selektierenden Principal Components durch die Größe der Eigenwerte vorgegeben. Für die gefundenen Independent Components existiert diese Reihenfolge nicht.

Es gibt deshalb verschiedene Ansätze, die ICA zur Dimensionsreduktion zu benutzen:

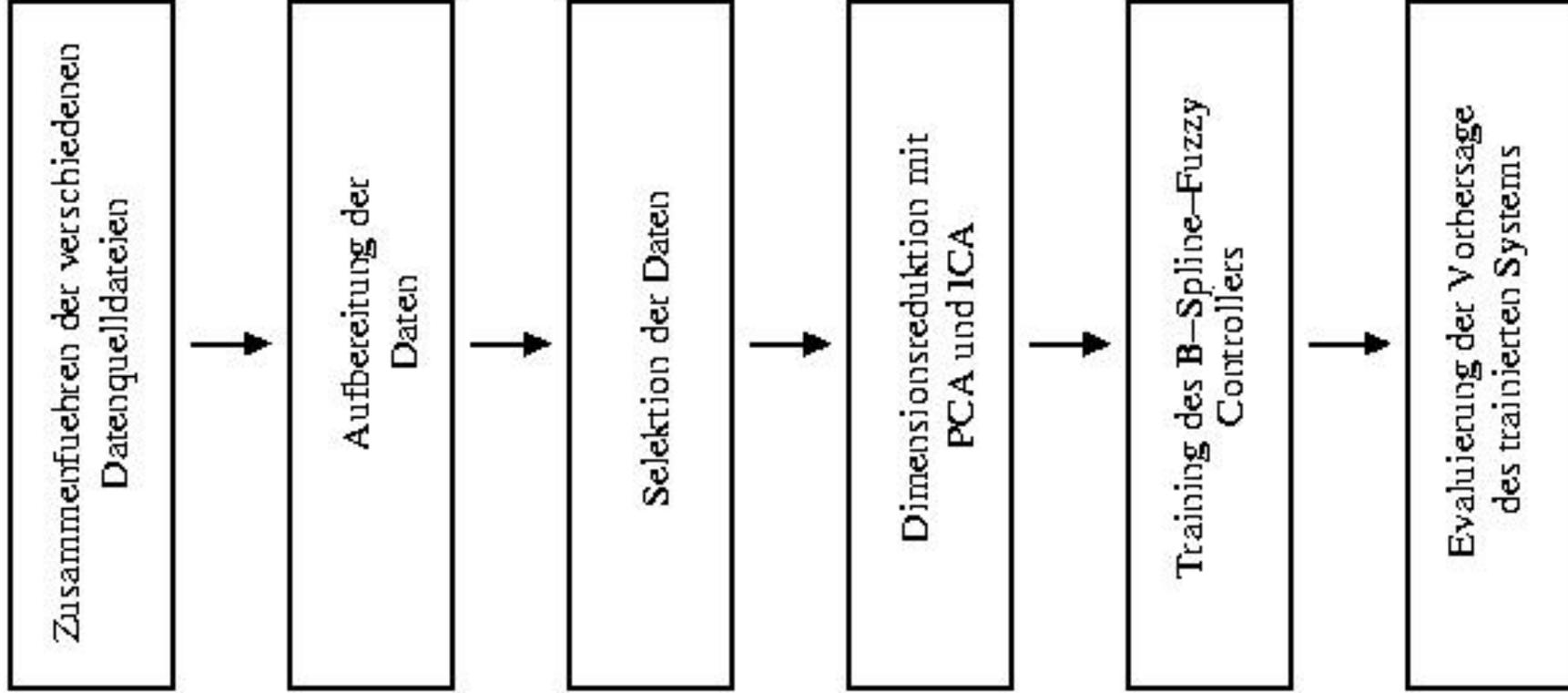
1. Ordnung der Zeilen in der Mischungsmatrix  $A$  nach der euklidischen  $L_2$ -Norm. Die Zeilen von  $A$  mit der größten  $L_2$ -Norm haben die größte Energie und somit haben die Quellen die zu diesen Zeilen gehören, einen größeren Einfluß auf die beobachteten gemischten Signale  $\mathbf{x}(t)$ .

2. Selektion der  $m$  Quellen mit der größten Amplitude, also der Komponenten  $s_i(t)$  der Vektoren  $\mathbf{s}(t)$  mit der größten  $L_\infty$ -Norm.
3. Eine weitere Möglichkeit ist die Berechnung der Independent Component Analysis auf Principal Components. Bei diesem Verfahren werden die  $n$ -dimensionalen Eingangsvektoren  $\mathbf{x}(t)$  mit Hilfe der PCA auf  $m$ -dimensionale Vektoren reduziert. Auf diesen reduzierten Vektoren wird dann die ICA berechnet.  
(Geschwindigkeitsvorteil).
4. Als weitere Möglichkeit bietet sich Input Selection an, um die Komponenten mit den interessantesten Informationen zu selektieren.

# Anwendungsgebiete ICA

- Filtern von MEG/EEG Daten
- Reduzierung von Rauschen in natürlichen Bildern
- Telekommunikation
- Auffinden von versteckten Faktoren in Finanzdaten

# Aufbau des Vorhersagesystems



# Vorhersage ohne fundamentale Daten

Bei der ersten Testreihe soll für zwei verschiedene Paare von Trainings- und Testmengen das unterschiedliche Verhalten des Vorhersagesystems für die unterschiedlichen Methoden der Dimensionsreduktion analysiert werden.

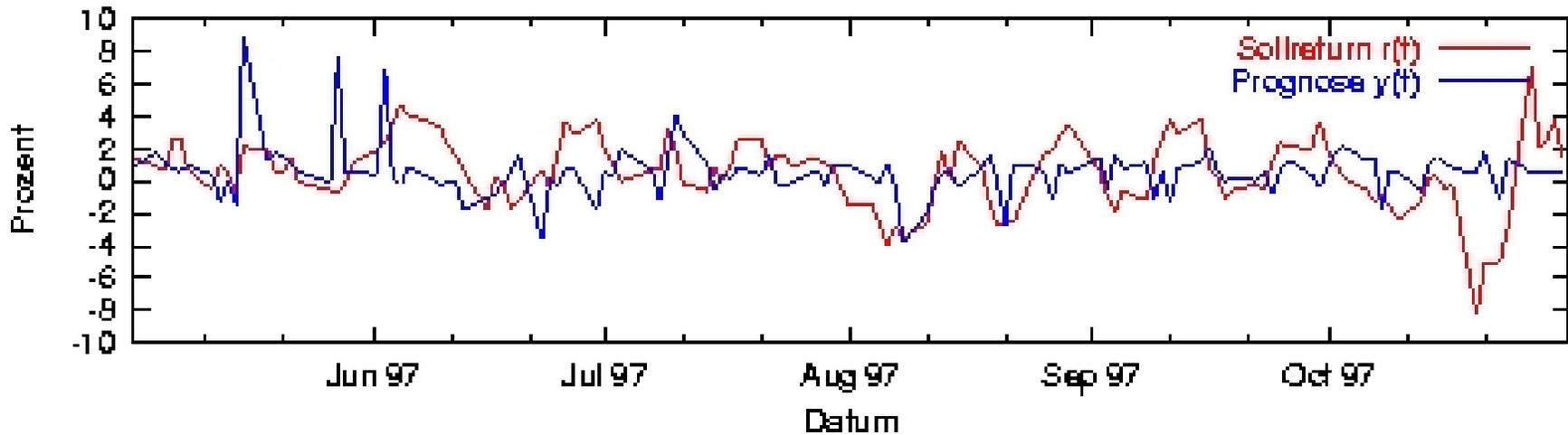
Der Vorhersagehorizont  $h$  ist 5 und die drei Paare von Trainings- und Testmengen sind Standardintervalle. Für jeden Tag wurde der Vektor  $\mathbf{x}(t)$  aus den Returns in Prozent  $r_h(t)$  für den S&P 500 Index wie in Gleichung 16 berechnet.

$$\mathbf{x}(t) = (r_{-1}(t), r_{-2}(t), \dots, r_{-50}(t))^T \quad (16)$$

Als Dimensionsreduktionsverfahren werden in dieser Testreihe die

PCA, die ICA mit der  $L_2$ -Norm als Selektionskriterium gegenübergestellt.

Die Sollreturns  $r(t)$  und die prognostizierten Returns  $y(t)$  für die Vorhersage mit der ICA:



# Vorhersage mit fundamentalen Daten

Mit dieser Methode soll herausgefunden werden, ob sich die Vorhersagen, basierend auf der Zeitreihenanalyse mit ICA, durch Hinzunahme der fundamentalen Daten verbessern lassen.

Der Vektor  $\mathbf{p}(t)$  besteht hier aus 6 mittels  $L_2$ -Norm berechneten Independent Components, die aus k-Day>Returns des S&P 500 Index berechnet wurden.

Als 7. Komponente kommt zusätzlich der jeweils aktuelle Wert aus einer der 10 Zeitserien  $f_i$  mit den fundamentalen Daten hinzu. Der Vorhersagehorizont  $h$  ist 5 und der Vektor  $\mathbf{x}(t)$  wie in Gleichung.

$$\mathbf{x}(t) = (r_{-1}(t), r_{-2}(t), \dots, r_{-50}(t), f_i(t))^T. \quad (17)$$

Der Vektor  $\mathbf{p}(t)$  wird mittels ICA aus den Return-Komponenten  $r_{-i}(t)$  des Vektors  $\mathbf{x}(t)$  berechnet. Die letzte Komponente  $f_i(t)$  geht direkt in den Vektor  $\mathbf{p}(t)$  ein.

In einer der folgenden Testreihen wird dann die Performance des Vorhersagesystems mit den Independent Components der Zeitreihenanalyse und den Kombinationen aller vielversprechenden Kandidaten untersucht.

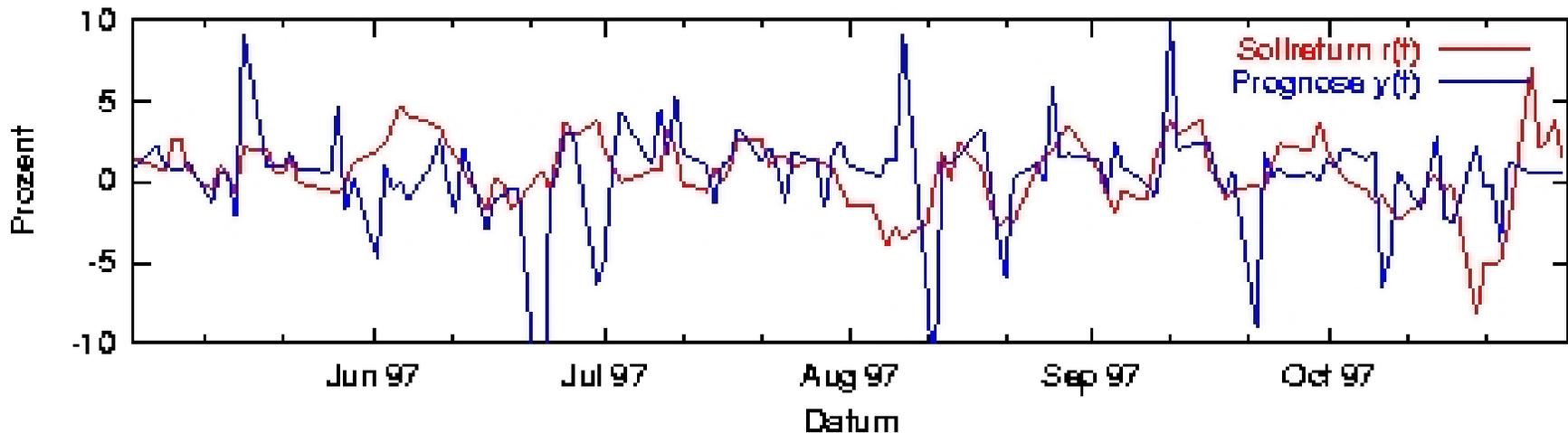
Für das Vorhersagesystem wurden US-amerikanische Indikatoren und Aktienindizes verwendet. Das System sollte sich aber auch auf andere Märkte übertragen lassen, sofern für die jeweiligen Märkte genügend Datenmaterial öffentlich zugänglich ist.

Das entwickelte Prognosemodell ermöglicht die Kombination von Zeitreihenanalyse mit der Analyse fundamentaler Daten. Die folgende Tabelle gibt eine Übersicht über die benutzten fundamentalen Daten:

Name	Kürzel
U.S. Weekly Leading Index	wli
U.S. Coincident Leading Index	usci
M3 Money Stocks	m3
Exchange Rate Swiss Franc/US\$	exszus
Consumer Price Index	cpiaucsl
Civilian Unemployment Rate	unrate
Consumer Sentiment Index	umcsent
Real Disposable Personal Income	dspic96
NAPM Manufact. Composite Index	napm
Manufacturers' New Orders	neworder

Überblick über die fundamentalen Daten

Die Geldmenge M3 als zusätzlicher Eingang hat die Vorhersage auf allen drei Intervallen verbessert. Die Indikatoren “Manufacturers New Orders” (neworder), “Arbeitslosenquote” (unrate) und “Real Disposal Personal Income” (dspic96) verbessern die Vorhersage bezüglich des “Mean Profit per Trade” Kriteriums auf jeweils zwei der drei Intervalle.



Die Sollreturns  $r(t)$  und die prognostizierten Returns  $y(t)$  für die Vorhersage aus 6 Independent Components und den Auftragseingängen (neworder) als zusätzlichem Eingang auf der Testmenge des Intervalls 2.